IN THE CLAIMS

1. (Presently Amended) A method of configuring software, the method comprising the steps of:

utilizing a common software configuration markup language in a computer system including at least one server which is connected to a plurality of client computers, the server having a memory for storing product elements which may be linked together to form at least one version of a software package, and further for storing processes usable by the clients to at least one of maintain the product elements and develop new product elements, and tools usable by the clients to at least one of define, maintain and update relations between product elements;

wherein the common software configuration markup language is ~~operable~~ useable to ~~at least one of~~ define a project, perform a memory access to at least one of product elements and packages, map one or more of the processes and tools to one ~~ore~~ or more library systems, and define relations between the product elements, and further wherein a framework associated with the common software configuration markup language is system-independent.

2. (Presently Amended) The method of claim 1, further comprising at least one of the steps of:

generating and storing a project definition which defines data flow, process flow and process definition of a project by using the common software configuration markup language;

generating and storing the software configuration framework by using the project definition and the process definition to describe the product elements and processes and define their access parameters and their relations;

mapping of the product elements, processes and tools to the one ~~ore~~ or more library systems by using the common software configuration markup language;

storing the product elements, processes and tools in the memory of the at least one server;

assigning the product elements, processes and tools in storage to the software configuration framework; and

calling selected ones of the product elements, tools and processes by at least one of the clients using commands of the common software configuration markup language, whereby another

programming language may be used to at least one of develop, update and test the contents of the product elements called.

3. (Original) The method of claim 1, wherein the common software configuration markup language is an extensible markup language.

4. (Presently Amended) The method of ~~to~~ claim 1, wherein the common software configuration markup language is based on the Extensible Markup Language (XML).

5. (Original) The method of claim 1, further comprising the step of generating commands in the common software configuration markup language via a control program in the server, wherein the control program is part of the software configuration framework.

6. (Original) The method of claim 1, wherein the common software configuration markup language is used for defining a project to one of develop and update a software product in a first server via the steps of accessing project definition data and data on project states, user access rights and element relations in a database connected to the first server to set up the project, and repeatedly accessing and updating project status data in the database.

7. (Original) The method of claim 1, wherein the common software configuration markup language is used for preparing an edit action for product elements comprising the steps of invoking an editor to which the product and process definition, element states and relations among elements are provided by a control program in a first server, wherein the control program is associated with the software configuration framework, and accessing processes and tools in the database for one of developing new product elements and modifying and updating existing product elements by the control program.

8. (Original) The method of claim 7, wherein the editor uses a language which is independent of the common software configuration markup language.

7

9. (Original) The method of claim 1, wherein the common software configuration markup language is used for building packages of product elements in a second server, connected to a database, via the steps of accessing processes and tools in one of a memory of the second server and the database so as to compile and link product elements to form program packages.

10. (Original) The method of claim 1, further comprising the step of generating self-documenting product elements and packages of product elements, and updates thereof, via the common software configuration markup language.

11. (Original) The method of claim 2, wherein the storing step comprises the step of storing the system-independent software configuration framework in memories of a plurality of geographically distributed computer systems which are connected together through a communication network, and wherein the assigning step comprises the step of assigning the product elements, processes and tools to the system-independent software configuration framework in the memories of each of the distributed computer systems.

12. (Original) The method of claim 11, wherein each of the distributed computer systems generates commands in the common software configuration markup language by a control program which is part of the software configuration framework.

13. (Original) The method of claim 11, further comprising the steps of editing product elements in each of the distributed systems by using a first server which is connected to a locally installed database, and building packages by a second server connected to the database.

14. (Original) The method of claim 11, wherein the calling step comprises calling elements, tools and processes by at least one of the clients attached to the servers in each of the distributed systems, and editing the elements by using one of same and different programming languages which are independent of the common software configuration markup language.

15. (Original) The method of claim 11, further comprising the step of transferring changes of the product elements, which result from processing of the product elements by each of the distributed systems, to the other distributed systems.

16. (Presently Amended) The method of claim 11, wherein one of the distributed systems is used as a master system, the memory of which contains project definition and process definition data in the common software configuration markup language, the data being used by a control program to initiate the software configuration framework.

17. (Presently Amended) Apparatus for configuring software, the apparatus comprising:

one or more processors operable to utilize a common software configuration markup language in a computer system including at least one server which is connected to a plurality of client computers, the server having a memory for storing product elements which may be linked together to form at least one version of a software package, and further for storing processes usable by the clients to at least one of maintain the product elements and develop new product elements, and tools usable by the clients to at least one of define, maintain and update relations between product elements;

wherein the common software configuration markup language is ~~operable~~ useable to ~~at least one of~~ define a project, perform a memory access to at least one of product elements and packages, map one or more of the processes and tools to one ~~ore~~ or more library systems, and define relations between the product elements, and further wherein a framework associated with the common software configuration markup language is system-independent.

18. (Presently Amended) The apparatus of claim 17, wherein the one or more processors are further operative to at least one of:

generate and store a project definition which defines data flow, process flow and process definition of a project by using the common software configuration markup language;

generate and store the software configuration framework by using the project definition and the process definition to describe the product elements and processes and define their access parameters and their relations;

map of the product elements, processes and tools to the one ~~ore~~ <u>or</u> more library systems by using the common software configuration markup language;

store the product elements, processes and tools in the memory of the at least one server;

assign the product elements, processes and tools in storage to the software configuration framework; and

call selected ones of the product elements, tools and processes by at least one of the clients using commands of the common software configuration markup language, whereby another programming language may be used to at least one of develop, update and test the contents of the product elements called.

19. (Original) The apparatus of claim 17, wherein the common software configuration markup language is an extensible markup language.

20. (Presently Amended) The apparatus of ~~to~~ claim 17, wherein the common software configuration markup language is based on the Extensible Markup Language (XML).

21. (Original) The apparatus of claim 17, wherein the one or more processors are further operative to generate commands in the common software configuration markup language via a control program in the server, wherein the control program is part of the software configuration framework.

22. (Original) The apparatus of claim 17, wherein the common software configuration markup language is used for defining a project to one of develop and update a software product in a first server via the steps of accessing project definition data and data on project states, user access rights and element relations in a database connected to the first server to set up the project, and repeatedly accessing and updating project status data in the database.

23. (Original) The apparatus of claim 17, wherein the common software configuration markup language is used for preparing an edit action for product elements comprising the steps of invoking an editor to which the product and process definition, element states and relations among elements are provided by a control program in a first server, wherein the control program is associated with the software configuration framework, and accessing processes and tools in the database for one of developing new product elements and modifying and updating existing product elements by the control program.

24. (Original) The apparatus of claim 23, wherein the editor uses a language which is independent of the common software configuration markup language.

25. (Original) The apparatus of claim 17, wherein the common software configuration markup language is used for building packages of product elements in a second server, connected to a database, via the operation of accessing processes and tools in one of a memory of the second server and the database so as to compile and link product elements to form program packages.

26. (Original) The apparatus of claim 17, wherein the one or more processors are further operative to generate self-documenting product elements and packages of product elements, and updates thereof, via the common software configuration markup language.

27. (Original) The apparatus of claim 18, wherein the storing operation comprises storing the system-independent software configuration framework in memories of a plurality of geographically distributed computer systems which are connected together through a communication network, and wherein the assigning operation comprises assigning the product elements, processes and tools to the system-independent software configuration framework in the memories of each of the distributed computer systems.

28. (Original) The apparatus of claim 27, wherein each of the distributed computer systems generates commands in the common software configuration markup language by a control program which is part of the software configuration framework.

29. (Original) The apparatus of claim 27, wherein the one or more processors are further operative to edit product elements in each of the distributed systems by using a first server which is connected to a locally installed database, and building packages by a second server connected to the database.

30. (Original) The apparatus of claim 27, wherein the calling operation comprises calling elements, tools and processes by at least one of the clients attached to the servers in each of the distributed systems, and editing the elements by using one of same and different programming languages which are independent of the common software configuration markup language.

31. (Original) The apparatus of claim 27, wherein the one or more processors are further operative to transfer changes of the product elements, which result from processing of the product elements by each of the distributed systems, to the other distributed systems.

32. (Original) The apparatus of claim 27, wherein one of the distributed systems is used as master system, the memory of which contains project definition and process definition data in the common software configuration markup language, the data being used by a control program to initiate the software configuration framework.

33. (Presently Amended) A software configuration system for the use in a computer system including at least one server which is connected to a plurality of client computers, the server having a memory for storing product elements which may be linked together to form at least one version of a software package, and further for storing processes usable by the clients to at least one of maintain the product elements and develop new product elements, and tools usable by the clients

to at least one of define, maintain and update relations between product elements, the system comprising:

means for storing a project definition which defines data flow, process flow and process definition of a project by using a common software configuration markup language which is adapted to serve as a meta syntax for defining a project and processes to perform the project, for mapping of the processes and tools to a database, and for accessing product elements in the database;

means for generating and storing a software configuration framework by using the project definition and the process definition to describe the product elements and processes and define their access parameters and their relations;

means for mapping of the product elements, processes and tools to the one ore or more library systems by using the common software configuration markup language;

means for storing the product elements, processes and tools in the memory of the at least one server;

means for assigning the product elements, processes and tools in storage to the software configuration framework; and

means for calling selected ones of the product elements, tools and processes by at least one of the clients using commands of the common software configuration markup language, whereby another programming language may be used to at least one of develop, update and test the contents of the product elements called.


34. (Presently Amended) An article of manufacture for configuring software, the article comprising a machine readable medium containing one or more programs which when executed implement the steps of:

utilizing a common software configuration markup language in a computer system including at least one server which is connected to a plurality of client computers, the server having a memory for storing product elements which may be linked together to form at least one version of a software package, and further for storing processes usable by the clients to at least one of maintain the product elements and develop new product elements, and tools usable by the clients to at least one of define, maintain and update relations between product elements;

13

wherein the common software configuration markup language is ~~operable~~ useable to ~~at least one of~~ define a project, perform a memory access to at least one of product elements and packages, map one or more of the processes and tools to one ~~ore~~ or more library systems, and define relations between the product elements, and further wherein a framework associated with the common software configuration markup language is system-independent.

35. (Presently Amended) The article of claim 34, further comprising at least one of the steps of:

generating and storing a project definition which defines data flow, process flow and process definition of a project by using the common software configuration markup language;

generating and storing the software configuration framework by using the project definition and the process definition to describe the product elements and processes and define their access parameters and their relations;

mapping of the product elements, processes and tools to the one ~~ore~~ or more library systems by using the common software configuration markup language;

storing the product elements, processes and tools in the memory of the at least one server;

assigning the product elements, processes and tools in storage to the software configuration framework; and

calling selected ones of the product elements, tools and processes by at least one of the clients using commands of the common software configuration markup language, whereby another programming language may be used to at least one of develop, update and test the contents of the product elements called.